2017-06

Investigating background pictures for picture gesture authentication

Monroy, Pauline

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INVESTIGATING BACKGROUND PICTURES FOR PICTURE GESTURE AUTHENTICATION

by

Pauline Monroy

June 2017

Thesis Co-Advisors: Paul C. Clark
Alan Shaffer

Reissued 3 Aug 2017 with corrections to order of in-text source citations.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	06-16-2017	Master's Thesis 06-30-2016 to 09-23-2016

4. TITLE AND SUBTITLE

INVESTIGATING BACKGROUND PICTURES FOR PICTURE GESTURE AUTHENTICATION

5. FUNDING NUMBERS

6. AUTHOR(S)

Pauline Monroy

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The military relies heavily on computer systems. Without a strong method of authentication to access these systems, threats to confidentiality, integrity, and availability of government information are likely to be more successful. A recent method of authentication for the Windows 8 and Windows 10 operating systems is picture gesture authentication (PGA), a new approach to entering a password to authenticate a user during system login. Each PGA password is composed of three gestures that are drawn over a picture chosen by the user. Strength requirements are set for PGA passwords similarly to text-based passwords. For simplicity, users tend to use shapes, colors, and objects in a picture, called points of interest (POI), as guidance when creating each gesture for their password. This concept provides an opportunity for potential hackers to make logical password guesses, decreasing the security of PGA. Previous work on PGA security used a proprietary brute-force algorithm to guess passwords based on POIs. We present a similar brute-force algorithm that is publicly available. We evaluate the efficiency of the new algorithm against various background pictures and propose strength requirements to improve the security of PGA.

14. SUBJECT TERMS	15. NUMBER OF PAGES
password, authentication, picture gesture authentication, background picture, strength requirements	61
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

INVESTIGATING BACKGROUND PICTURES FOR PICTURE GESTURE AUTHENTICATION

Pauline Monroy
Civilian, Department of the Navy
B.S., California State University Monterey Bay, 2014

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 2017

Approved by: Paul C. Clark
Thesis Co-Advisor

Alan Shaffer
Thesis Co-Advisor

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The military relies heavily on computer systems. Without a strong method of authentication to access these systems, threats to confidentiality, integrity, and availability of government information are likely to be more successful. A recent method of authentication for the Windows 8 and Windows 10 operating systems is picture gesture authentication (PGA), a new approach to entering a password to authenticate a user during system login. Each PGA password is composed of three gestures that are drawn over a picture chosen by the user. Strength requirements are set for PGA passwords similarly to text-based passwords. For simplicity, users tend to use shapes, colors, and objects in a picture, called points of interest (POI), as guidance when creating each gesture for their password. This concept provides an opportunity for potential hackers to make logical password guesses, decreasing the security of PGA. Previous work on PGA security used a proprietary brute-force algorithm to guess passwords based on POIs. We present a similar brute-force algorithm that is publicly available. We evaluate the efficiency of the new algorithm against various background pictures and propose strength requirements to improve the security of PGA.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

BDAS	background draw a secret
DAS	draw a secret
DOD	Department of Defense
PGA	picture gesture authentication
PII	personally identifiable information
POI	point of interest

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

Thank you to Zhao et al. for providing the data necessary to reimplement your algorithm to complete my thesis. Partial support for this work was provided by the National Science Foundation's CyberCorps: Scholarship for Service (SFS) program under Award No. 1241432. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Most importantly, thank you to my advisors, Professor Paul Clark and Dr. Alan Shaffer. They are very kind, patient, helpful and quick to provide feedback. You were both my favorite professors before becoming my advisors and I am happy to have the chance to work with you!

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Motivation

The use of passwords as a method of authenticating someone's claim of identity dates back to ancient times in the Roman military in which Romans referred to passwords as "watchwords" [1]. Since then, passwords have been modified to what we have today. Traditional computer-based authentication methods use text-based passwords, which are a string of alphanumeric characters and symbols used to authenticate a user before granting that user access to a device or program. For security purposes, many programs use strength requirements for passwords. Strength requirements may include a certain number of uppercase alphabetic characters, lowercase alphabetic characters, symbolic characters, or numerical characters, and a minimum and maximum length. They may also require a password to be changed after a period of time, and that no repeated passwords may be used. Even with these strength requirements, there remain weaknesses in text-based passwords.

Suo et al. said "human factors are often considered the weakest link in a computer security system" [2]. Zhao et al. found that people use simple passwords because they are easier to remember [3], [4]. Therefore, dictionary attacks were created, where a list of plausible passwords are generated based on dictionary words, and used to guess passwords. Other human factors related to text-based passwords include users recycling passwords throughout different programs or re-using passwords for the same program. Users also tend to write down their password, either on a sticky note left on their desk or in an unencrypted document on their system. In either case, if the password is found there can be numerous consequences to security. If a password gets in the wrong hands, it can lead to illicit access onto a private network, or a data breach.

Since text-based passwords are difficult for people to keep track of, other methods of authentication have been developed.Suo et al. believe that people are more likely to remember a visual password [2]. Picture gesture authentication (PGA) is a new type of authentication that uses picture-based passwords, and is the scope of this thesis. In particular, the research

looks at the best types of background pictures for more secure PGA.

We proceed by presenting a brute-force algorithm, designed after the work of Zhao et al. [3], [4], that makes logical guesses to crack the PGA password of a user given a specific picture. We programmed the algorithm to use points of interest (POI), which are specific areas of a picture that may catch the eye of a user, to determine likely choices of a password. By analyzing the accuracy and efficiency of the algorithm to generate brute-force passwords, we determine the variety of pictures that are superior for a background picture. We show that the background picture selected can increase the strength of the password chosen for PGA.

1.2 Benefits to the Navy

The main contribution of this research is to investigate the security bounds of picture gesture authentication. The Navy would benefit from this study because if PGA is not a strong method of authentication, then potential threats to confidentiality, integrity, and availability of government information are plausible. Strong authentication is recommended by the DOD Cybersecurity Discipline Implementation Plan that was amended February 2016.

> Reducing anonymity as well as enforcing authenticity and accountability for actions on DOD information networks improves the security posture of the DOD. The connection between weak authentication and account takeover is well-established. Strong authentication helps prevent unauthorized access, including wide-scale network compromise by impersonating privileged administrators. Commanders and Supervisors will focus attention on protecting high-value assets, such as servers and routers, and privileged system administrator access. This line of effort supports objective 3-4 in the DOD Cyber Strategy, requiring the DOD CIO to mitigate known vulnerabilities by the end of 2016. [6]

An agreement between Microsoft and the DOD provides the Navy with the newest versions of Microsoft products, including Windows 8 and 10, which both use PGA. Navy Rear Admiral David G. Simpson, DISA's vice director and senior procurement executive explained that the DOD has continued to focus on mobile computing, stating "Microsoft is committed to making sure that the technology within the agreement has a mobile-first focus, and we

expect to begin to take advantage of Microsoft's mobile offerings as part of our enterprise mobility ecosystem" [2]. Microsoft claims that PGA passwords are more secure than text-based passwords [5], and that DOD users will be more likely to use PGA. It is important, however, that PGA not be used in an insecure fashion, therefore, this study is important to help the Navy make the best decision on background pictures for the security of PGA.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, we describe the history, notation, and brute-forcing of PGA, as well as related work. Chapter 3 discusses the two corpora used to test the program created for this thesis. The process of POI extractions and functions used for the brute-force algorithm are covered in Chapter 4. In Chapter 5, the passwords, POIs, and results of the program for each picture are explained. Finally, the conclusions and recommendations, and suggested future work of this thesis are presented in Chapter 6.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background and Related Work

In this chapter, we explain the process of using picture gesture authentication (PGA) on a Windows 8 device. A key insight is how users tend to select points of interest (POI) to choose the location of gestures. POIs are a key concept employed by prior work on brute-forcing a password under PGA. We also summarize related work on picture authentication schemes. For clarity and ease of comparison, we adopt the notation of Zhao et al. [3], [4].

2.1 Picture Gesture Authentication

Authentication is any mechanism used to validate if someone is the identity they claim to be on a computer system or program. There are three broad approaches to authentication, often referred to as something you know, something you own, or something you are. PGA is a relatively new authentication mechanism that falls under the umbrella of something you know. Microsoft started using PGA as an optional replacement for text passwords with their Windows 8 consumer technology. This new method of authentication was announced by Microsoft in late 2011 [5] for all versions of Windows 8 and products supporting PGA as a primary method of authentication were released on October 26, 2012.

With the Windows 8 operating system, by default, user accounts are configured to use text-based passwords. To use PGA, the user selects the picture password sign-in option. After providing proper credentials, the user is required to choose a picture from their picture library. Using their own picture, instead of Microsoft providing one, will increase the security of PGA. The intuition is that two users are likely to select different pictures, as PGA is configurable per-user. After a picture is chosen, the user is prompted to create a password. A password for picture gesture authentication (PGA) is a series of gestures, limited to taps, circles or lines drawn on the picture. The users are expected to draw three gestures on the picture using their finger or stylus on the touchscreen or a mouse if no touchscreen is available. When users later authenticate with PGA, they must redraw the selected gestures, in the original order, on their chosen picture.

We record a gesture password as a sequence of three gestures, $\vec{\pi} = \pi_1\pi_2\pi_3$. Each $\vec{\pi}$ is

one of many passwords in the password space, $\vec{\pi} \in \vec{\prod}$. Each gesture in the password is represented as a 7-tuple: $\pi_i = \langle g, x_1, y_1, x_2, y_2, r, d \rangle$. Let $g \in \{tap, circle, line\}$ be the type of gesture. The first coordinate (x_1, y_1) can indicate a tap point, the center of a circle or the first point of a line. The second coordinate (x_2, y_2) represents the end of a line, and is unused for other gesture types (i.e., let $(x_2, y_2) = (0, 0)$ for a tap or circle). Let r be the radius of a circle gesture, and otherwise unused (i.e., $r = 0$ for a line or tap). Let $d \in \{+, -, 0\}$ be the direction in which a circle is drawn, indicating a clockwise or counterclockwise gesture, and otherwise unused (i.e., 0 for a tap or line). Each gesture is one of many possible gestures in the gesture space, $\pi_i \in \prod$.

Figure 2.1 shows an example gesture password. The first gesture, $\pi_1 = \langle circle, 35, 15, 0, 0, 9, - \rangle$, is a counterclockwise circle around the man's head centered at $(35, 15)$ with a radius of 9. The second gesture, $\pi_2 = \langle line, 54, 34, 79, 27, 0, 0 \rangle$, is a line from $(54, 34)$ to $(79, 27)$, from one woman's nose to another's. The last gesture, $\pi_3 = \langle tap, 16, 35, 0, 0, 0, 0 \rangle$, is a tap on the left woman's nose, at coordinate point $(16, 35)$.

Figure 2.1. Example of a Sequence of Gestures on a Picture. Adapted from [3], [4].

Naturally, human error is likely to occur when redrawing passwords. Therefore, a distance

function is built into the authentication process. Since pictures come in various sizes, the longest dimension is divided into 100 and the shortest dimension is scaled accordingly [5]. The pictures are scaled to determine the coordinate points that fall within an error distance of the actual coordinate point used for a gesture. When entering a password, if a coordinate point of a gesture is within the error distance of the actual coordinate point, that point will be accepted.

Figure 2.2 shows an example of the points accepted during authentication within a distance of 3 around the recorded point of the actual password [5]. All of the gesture points within 3 of the actual gesture point, shaded in green, are at least 90% accurate to the actual point within the error distance, and would be accepted during user login. The yellow, orange, and red points are not close enough to the actual gesture point to be accepted during user login. For example, a tap on $(14, 35)$ would suffice for the gesture $\pi_3 = \langle tap, 16, 35, 0, 0, 0, 0 \rangle$ since the distance $d((16, 35), (14, 35)) \leq 2\sqrt{3}$.

70%	77%	82%	85%	86%	85%	82%	77%	70%
77%	84%	89%	92%	93%	92%	89%	84%	77%
82%	89%	94%	97%	98%	97%	94%	89%	82%
85%	92%	97%	100%	100%	100%	97%	92%	85%
86%	93%	98%	100%	100%	100%	98%	93%	86%
85%	92%	97%	100%	100%	100%	97%	92%	85%
82%	89%	94%	97%	98%	97%	94%	89%	82%
77%	84%	89%	92%	93%	92%	89%	84%	77%
70%	77%	82%	85%	86%	85%	82%	77%	70%

Figure 2.2. Points $\leq 90\%$ to the 100% Exact Matched Point Are Accepted During Authentication. Adapted from [5].

2.2 Brute-Forcing PGA

Zhao et al. [3], [4] provide the intuition that users select gestures by employing points of interest (POIs) embedded in the underlying picture. POIs can be described by many features, such as objects $\mathscr{D}_o = \{head, eye, mouth, nose, bike, dog, \dots\}$, colors $\mathscr{D}_c = \{blue, red, yellow, green, \dots\}$, shapes $\mathscr{D}_s = \{square, circle, triangle, rectangle, \dots\}$ and other miscellaneous types, $\mathscr{D}_*$. These form an attribute space $D \subseteq 2^{\mathscr{D}}$ such that $\mathscr{D} = \mathscr{D}_o \bigcup \mathscr{D}_c \bigcup \mathscr{D}_s \bigcup \mathscr{D}_*$. Each POI is recorded as a 5-tuple, $\theta_k = \langle x_1, y_1, x_2, y_2, D \rangle$, which defines the POI in picture k that is enclosed by a rectangle bounded by coordinates (x_1, y_1) and (x_2, y_2), and has the set of attributes $\mathscr{D}$.

Referring back to the working example in Figure 2.1, the POIs include the heads of each person, their eyes, their noses, their mouths, the linear edge of the curtain, the blue lines in the man's shirt, the black dots on the girl's shirt, the woman's necklace, and the corner of the vanity. Just as users tend to select dictionary words for text passwords, it is believed that they tend toward POIs on a picture to choose their PGA passwords.

POIs help a user remember where they placed their gestures. This insight is used by Zhao et al. to provide an attack on PGA, comparable to a dictionary attack against text-based passwords. As mentioned in Section 2.1, it is unlikely that any two users would use the same picture for authentication. The attack framework requires previously seen passwords on known pictures to learn password-selection patterns to create a dictionary of gesture passwords. Machine analysis can then be used to identify POIs on pictures as a "dictionary" to guess a PGA password. This process is discussed in more detail in Chapter 3.

2.3 Related Work

There has been growing interest in providing an alternative to text passwords by using graphics. It has been argued that graphical passwords are more secure than text passwords, however, in "Graphical Passwords: A Survey," Suo et al. explain how brute-force attacks, dictionary attacks, guessing, spyware, shoulder surfing, and social engineering are used to attack graphical passwords, just like text passwords [2]. They claim the defense against graphical passwords is more difficult since N length text passwords have 94^N possible passwords based on 94 printable characters. On the other hand, PGA has only 1,155,509,083 possible passwords with three gestures, based on all the possible sets of three gestures made

by taps, circles, and lines [5] whish is less than 94^6, (the number of 6-character passwords). After guessing a graphical password, a program must be written to precisely draw such gestures on a picture. Suo et al. claimed in 2005 that there was no method of dictionary attacks on graphical passwords. Since then, research has shown that dictionary attacks are possible but must be designed for each individual picture, as described by Zhao et al. In Chapter 4, we explain how it is easy to guess graphical passwords since they are more predictable than text passwords. In 2013, Damopoulos et al. proved that there exists a touchlogger, similar to a keylogger but for touch screens, that can record gestures on touch screen devices [7]. This is a finding made after Suo et al. stated that spyware was unable to track picture passwords. This is important to keep in mind since PGA is often used on, though is not limited to, touch screen devices. Picture passwords are vulnerable to shoulder surfing as we will discuss more in this section. Picture passwords are said to be insusceptible to social engineering because it is difficult to explain to someone verbally how to recreate a password [2].

One of the vulnerabilities of text passwords is that users tend to recycle passwords for separate accounts because it is difficult to remember multiple strong passwords. Suo et al., however, affirm that there is no "convincing evidence" that picture passwords are easier than text passwords to memorize. De Luca et al. also conclude that authentication methods other than text-based passwords and personal identification numbers (PIN) should be used, after analyzing password pattern authentication [8]. Pattern passwords consist of a series of continuous edges made on a $3x3$ grid of points. They surveyed users over a period of time to collect data and study the passwords the users created, along with how they created them. Each user, they concluded, has a unique way of making each stroke. If used correctly, this pattern matching can be an additional method of authentication. Assuming an attacker knows the shape of the password, they may not be able to imitate the user's stroke motions, which falls under the *something you are* category of authentication.

Draw a Secret (DAS) is a picture-based password authentication method that allows a user to make a drawing on a blank grid as a password. This is different than PGA since there is not a background picture with POIs to guide users in creating a password, but is similar in the sense that a PGA password can be a picture drawn on a grid comprised of three gesture elements. Nali and Thorpe prove this scheme is insecure by showing that users center their drawings and use symmetry [9], [10]. Essentially, they show that this approach increases

the chances of guessing a password. Dunphy and Yan attempted to enhance this method of authentication by providing a background picture for a user as a guide to improve how they originally created passwords [11]. This scheme is called Background Draw a Secret (BDAS). This relates to PGA since they both have a background picture that directs users in constructing a password, but BDAS has less restrictions on the number and types of gestures used for creating a password. They found that BDAS closely relates to PGA since users are likely to use POIs. Since BDAS is like PGA, and PGA is insecure, therefore BDAS is insecure.

PassPoints is an authentication method that allows a user to choose points on a picture as a password. This is essentially a subset of the password space of PGA, with only the tap gesture being allowed. PassPoints is similar to DAS, containing a less structured password space to PGA, but when selecting passwords PGA has fewer rules than PassPoints. Wiedenbeck et al. studied the security of PassPoints and found that users tend to use taps corresponding to POIs, which they call "hotspots," when choosing points that correspond to POIs. The main outcome of their work is the recommendation that users should select pictures that avoid hotspots [12], [13].

Wiedenbeck et al. also found that users rely on POIs to assist in building passwords. Using the same dataset as Zhao et al. shown in Section 3.1, Alshehri et al. explored security of PGA, restricted to pictures with a high number of POIs. Since POIs are used to brute-force PGA, a background picture with more POIs would represent a larger password space, and thus provide more security against brute-force. As yet unpublished, they are developing a metric to find if a picture is suitably complex by validating pictures with more POIs to be less resistant to dictionary attacks [14]. Pictures with few POIs are more susceptible to attacks. Thus, Alshehri et al. claim there should be strength requirements of the background picture. In contrast, we are concerned with revalidating the premises and results of the original study by Zhao et al.

Most PGA methods are used with touch screen devices. In addition to click points, as mentioned by Alshehri et al., Aviv et al. found that smudge marks can be used to guess the passwords of any of the aforementioned types of picture authentication [15].

Picture password mechanisms are also susceptible to shoulder surfing. Logging in with PGA allows someone close by to easily see a user's password. To provide more security,

a system such as LatentGesture can help keep the PGA password more secure [16]. LatentGesture records a user's behavior on a touchscreen device such as the speed of swiping across the screen or typing patterns. These recorded behaviours build a model of that user. When it suspects the current user does not match the model, LatentGesture will automatically log off the system. Saravana described a study using 20 people that were asked to check boxes, swipe sliding bars, and tap buttons to fill out a form. With high accuracy, LatentGesture was able to identify the users correctly [17]. This is not a surprising result because LatentGesture combines the *something you know* authentication category with the *something you are* category.

Overall, picture gesture authentication has its weaknesses and vulnerabilities just like text-based passwords. Thus, we created a brute-force algorithm described in Chapter 4 to compare the security of one picture to another, determining the best selection of background pictures for an increase in security for PGA. Before describing the algorithm, we will discuss the data given by Zhao et al. that we have also used in our study.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Corpora

In this chapter, we discuss the data gathered and analyzed by Zhao et al. in their study [4]. Two corpora were employed in their study, both containing pictures and passwords created by the study's subjects. The Arizona-Turk dataset was an artificial dataset, where subjects generated passwords for a small set of images. The Arizona-Student dataset was a more authentic dataset, where university students generated personal passwords used by a website. The next two sections summarize the demographics of the subjects and the contents of the corpora in the study.

3.1 Arizona-Turk Dataset

The Arizona-Turk dataset (called *dataset 2* in the Zhao et al. study [3], [4]) was solicited by advertisements in the schools of engineering and business at two different universities, and gathered using Amazon's Mechanical Turk crowdsourcing service. Only individuals with previous security-related research experience were qualified to participate so they could understand the importance of this study.

In the Arizona-Turk dataset, 762 subjects were given 15 pictures (see Figure 3.1) drawn from the PASCAL Visual Object Classes Challenge 2007 dataset [18]. The subjects were prompted to pretend the pictures were protecting their bank information, with the intention of influencing subjects to make strong passwords for each of the 15 pictures. Not all subjects completed the entire task, so the number of passwords gathered for each picture is not the same (see Figure 3.2). A total of 10,039 passwords were gathered: on average, 669 passwords per picture and 13 passwords per subject. Interestingly, there were passwords which one might guess, such as circling tires on a bike and tapping a person's nose. Further discussion can be found in Chapter 4.

The subjects were given a demographic survey. Of the 762 subjects, only 652 (85.5%) filled out the survey. Of the 652 surveyed, 420 (64.4%) of them reported being male, 232 (35.6%) female; 243 (37.2%) were between 18 and 24 years of age, 296 (45.4%) between 25 and 34 years of age, and 98 (15%) between 35 and 50 years of age.

(a) 000243.jpg (b) 000316.jpg (c) 001116.jpg (d) 001358.jpg (e) 002057.jpg

(f) 002080.jpg (g) 002840.jpg (h) 003026.jpg (i) 003731.jpg (j) 004054.jpg

(k) 005570.jpg (l) 006412.jpg (m) 006467.jpg (n) 007628.jpg (o) 009899.jpg

Figure 3.1. The 15 Pictures from the Arizona-Turk Dataset. Source: [3], [4].

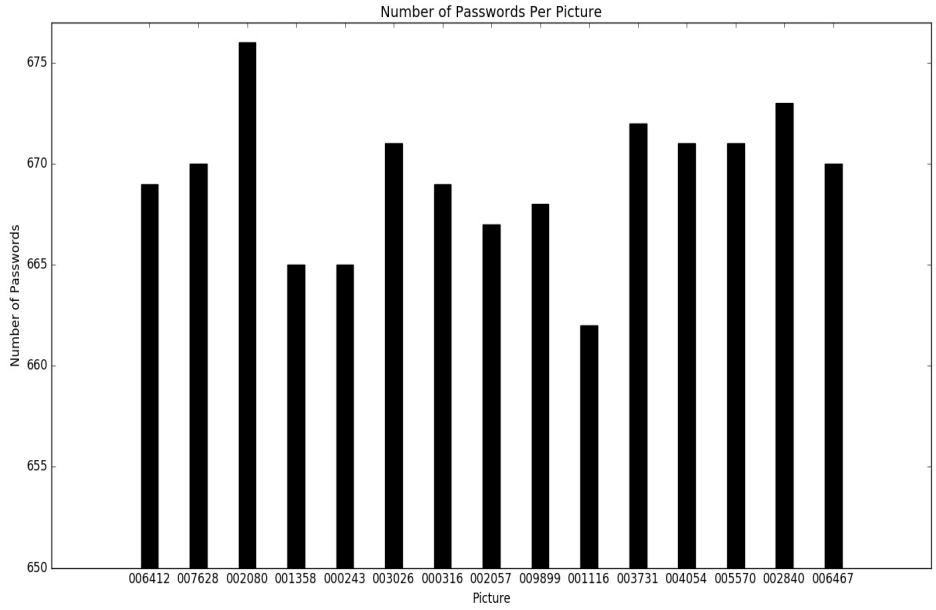


Figure 3.2. Number of Passwords for Each of the 15 Pictures in the Arizona-Turk Dataset

As part of the survey, a multiple choice question was asked to help understand the choice of passwords made by the subjects as follows: "Which of the following best describes what you are considering when you choose locations to perform gestures?" Of the subjects in this study, 389 (59.6%) answered, "I try to find locations where special objects are"; 143 (21.9%) answered, "I try to find locations where some special shapes are"; 57 (8.7%) answered, "I try to find locations where colors are different from their surroundings"; and 66 (10.1%) answered, "I randomly choose a location to draw without thinking about the background picture." Thus, 90.2% of respondents admitted to using a strategy of selecting POIs, which effectively limited the password space and, perhaps, biased it toward a POI populated area of the picture.

3.2 Arizona-Student Dataset

The Arizona-Student dataset (called *dataset 1* in the Zhao et al. study) was gathered from university students in a classroom setting. An authentication method modeled after PGA in Windows 8 was created to gather information on how students in an undergraduate computer science class would create passwords. This authentication method was used by the students to access the course website, containing class materials such as homework, assignments, grades, and lecture notes. Data was gathered over one semester, or approximately three and a half months.

The publicly released dataset contains subject IDs, a hash value for the picture, a password, and an activity log. The log recorded setting of passwords, attempted logins, the number of successful login attempts, and any password changes or new picture selections. Since students selected their own pictures, some contained family photos and other personally identifiable information (PII), so no pictures were released with the dataset.

A total of 56 students in the computer science class participated in the study. The data collected reflected: 69 different pictures,[1] 86 unique passwords, 2,536 login attempts (2,109 successful, 427 failed) and 172 registrations (86 registered, 86 confirmations). On average, each student used 2.5 pictures, made 37.66 successful login attempts, had 7.625 failed login attempts, registered 1.53 logins, and confirmed 1.52 logins (see Figure 3.3). Between the

[1] According to Zhao et al. [3], [4], there were 58 unique pictures; this does not match the calculations made with the public-released data.

registrations, confirmations, and successful and failed logins, there were a total of 2,708 datapoints.

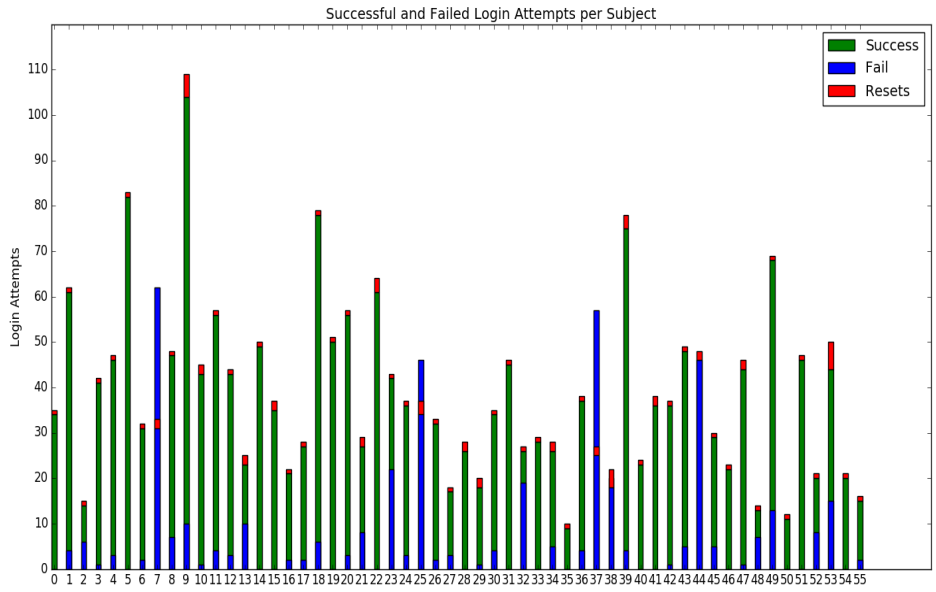


Figure 3.3. Number of Successful/Failed Login Attempts and Number of Reset Passwords per Subject in the Arizona-Student Dataset

The students were also asked the same demographic survey. Of the 56 students, only 33 (58.9%) filled out the survey. Of the 33 surveyed, 27 (81.8%) reported being male, and 6 (18.2%) female; 21 (63.6%) were between 18 and 24 years of age. Since the students were in an undergraduate course in computer science, it is reasonable that the numbers were not as diverse as those for dataset 2.

As in the Arizona-Turk dataset, the subjects of the Arizona-Student dataset were also asked the question, "Which of the following best describes what you are considering when you choose locations to perform gestures?" Of the 33 respondents, 24 (72.7%) answered, "I try to find locations where special objects are"; 8 (24.2%) answered, "I try to find locations where some special shapes are"; 0 (0%) answered, "I try to find locations where colors are different from their surroundings"; and 1 (3%) answered, "I randomly choose a location to draw without thinking about the background picture." Since students were asked to use this password to protect their actual course material, and to select their own pictures, we

expect that this dataset was more realistic than the Arizona-Turk dataset. This reflects an even stronger trend toward biased password selection.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
BestCover Algorithm

This chapter presents the *BestCover* algorithm described by Zhao et al. [4] to make a logical guess of an unknown password using a previously unseen picture. In Section 4.1, we detail the POIs in the Arizona-Student dataset and the Arizona-Turk dataset. In Section, 4.2 we define location dependent gesture selection functions and how they are used to map POIs into potential passwords for a picture. In Section 4.3, we then explain the *BestCover* algorithm, which uses a subset of the dataset for training and is evaluated on the remainder of the dataset. This is the same methodology employed by Zhao et al. to evaluate this algorithm [3], [4]. We adopt the notation of Zhao et al. for ease of comparison between our independent re-implementation and their original work.

4.1 POI Extraction

For each of the datasets in the Arizona case study, Zhao et al. extracted POIs with "mature computer vision techniques such as object detection, feature detection and objectness measure" [3], [4]. The POI attributes were categorized as follows: face, body, eye, ear, mouth, nose, head/shoulder, clock, airplane, unknown object, forehead, car, line type, circle type, color type, "no semantics" and "not valid."

The number of POIs extracted from the pictures in the Arizona-Student dataset are expressed in Figure 4.1. The number of POIs per picture varied widely between the pictures the students chose. Recall that for this dataset, some pictures were not made available due to PII concerns, however Zhao et al. [3], [4] provided information about the POIs (their type and their coordinate location on the picture). This eliminated the need to extract POIs using computer vision methods, and thus reduced many variables in the attempt to recreate an algorithm simliar to that of Zhao et al. to decide which background pictures are best to use in PGA. For the Arizona-Turk dataset, Figure 4.2 shows the number of POIs extracted for each of the 15 pictures in Figure 3.1. We observed a correlation between the variation in the number of POIs per picture, and the level of difficulty to brute force PGA passwords, described further in Chapter 5.

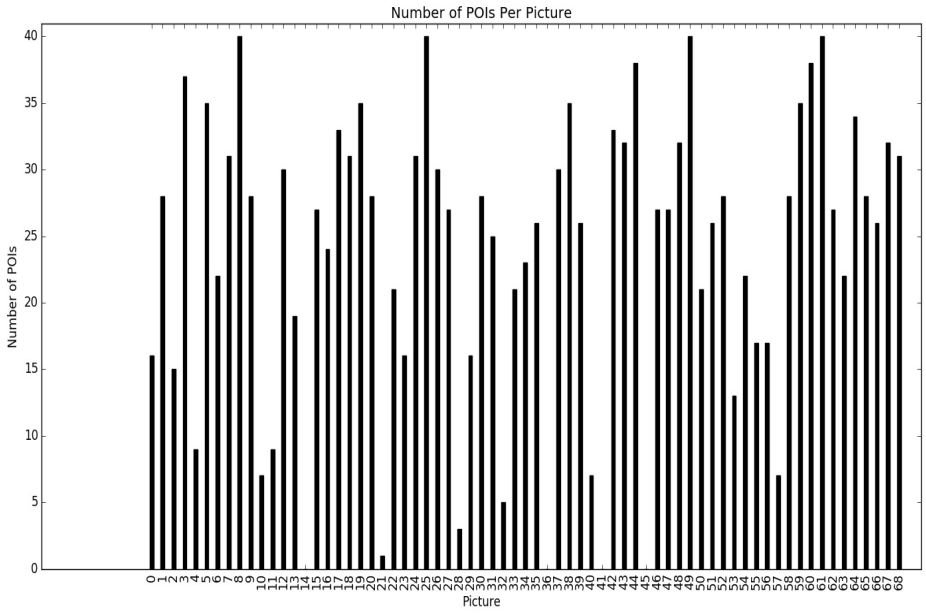


Figure 4.1. Number of POIs Extracted from the 58 Pictures in the Arizona-Student Dataset

4.2 Location Dependent Gesture Selection Functions

Users are likely to choose POIs on a picture when selecting a password. Therefore, mappings were created to aid the brute force method described in Section 4.3. Location dependent gesture selection functions (LdGSF) [3], [4] are mappings $s : G \times 2^{\mathscr{D}} \times 2^{\mathscr{D}} \times \Theta \rightarrow 2^{\Pi}$ from descriptions of gestures on POIs to PGA passwords using actual coordinate points of those POIs in the picture. The domain is the cross product of the set of gestures, the set of attributes at the first point, the set of attributes at the second point if the gesture is a line, and the set of POIs in the given picture, respectively. The range is the password space. Using the POIs extracted from the picture, as described in Section 4.1, a mapping can be made to describe gestures on a picture. A sequence of three LdGSF mappings, $\vec{s} = s_1 s_2 s_3$, will yield three gestures, making plausible passwords.

For example, referring to Figure 2.1 with the password $\vec{\pi} = \pi_1\pi_2\pi_3$, where $\pi_1 = \langle circle, 35, 15, 0, 0, 9, - \rangle$, $\pi_2 = \langle line, 54, 34, 79, 27, 0, 0 \rangle$, and $\pi_3 = \langle tap, 16, 35, 0, 0, 0, 0 \rangle$, the LdGSFs for the k^{th} picture p_k would be: $s_1 = s(circle, \{head\}, \emptyset, \theta_k)$, $s_2 = s(line, \{nose\}, \{nose\}, \theta_k)$, $s_3 = s(tap, \{nose\}, \emptyset, \theta_k)$.

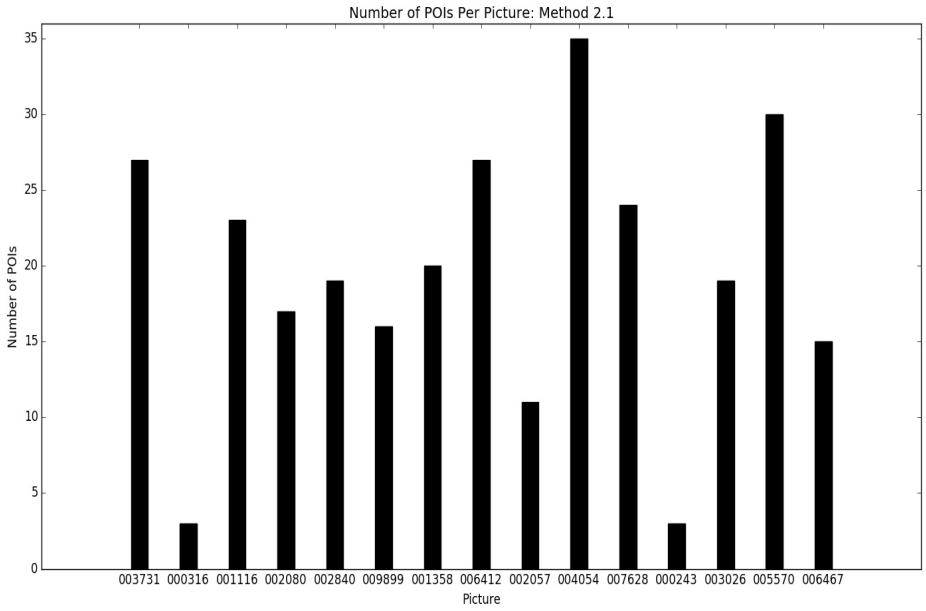


Figure 4.2. Number of POIs Extracted from the 15 Pictures in the Arizona-Turk Dataset

An LdGSF sequence can map to several passwords. For example, given the gesture made by $s_1 = s(circle, \{head\}, \emptyset, \theta_k)$ above, if a user decides to perform this gesture on Figure 2.1, there are four possible heads to circle and each circle can have a different circumference. Therefore, one LdGSF can produce many possible gestures, and a single LdGSF sequence can produce many possible passwords.

4.3 Brute-Force Algorithm

Since POIs on a picture may decrease a user's password space by steering them toward specific gestures, a brute force algorithm centered around this notion will assist in attacking a password for a previously unseen picture. Zhao et al. describe the *BestCover* algorithm to create a LdGSF sequence dictionary. The program written for this algorithm was not released to the public. Hence, we attempted to recreate their algorithm using known passwords to derive patterns of data that were used to prioritize guesses, providing the most efficient coverage of the password space, i.e., guesses were ordered by popularity of the relationship between POI and gesture. Figure 4.3, expresses in pseudocode the *BestCover*

algorithm in a way that aligns with our implementation of the original work of Zhao et al.

```
1:  function BestCover((s⃗_1, ..., s⃗_n), (π⃗_1, ..., π⃗_n))
2:      for s⃗_i in (s⃗_1, ..., s⃗_n) do
3:          for π⃗_j in (π⃗_1, ..., π⃗_n) do
4:              if π⃗_j ∈ s⃗_i then
5:                  s_i count + +
6:              end if
7:          end for
8:      end for
9:      for s⃗_i in (s⃗_1, ..., s⃗_n) do
10:         if s_i count ≠ 0 then
11:             S⃗' ← {s⃗_i : s⃗_i count}
12:         end if
13:     end for
14:     order ← sort S⃗' by max s⃗_i count
15:     return order
16: end function
```

Figure 4.3. The Pseudocode of $BestCover$. Adapted from [3], [4].

First, the LdGSF sequences were created separately. Each set of attributes collected for the LdGSFs was built from known passwords. Since the passwords contain coordinate points, if a point fell within an interval of a POI's location then that attribute and its gesture were recorded as an LdGSF. If the coordinate point fell within an intersection of multiple POIs then multiple attributes were added in the LdGSF.

The input to the $BestCover$ algorithm consists of the training data's LdGSF sequences and their corresponding passwords. Lines 2 – 5 verify the number of passwords that the LdGSF sequences produce from the training data, assigning them each a rating. The LdGSF sequences not found to produce any of the passwords are not beneficial to the final dictionary to produce passwords. In lines 9 – 11, only the LdGSF sequences with a ranking greater than zero are taken into consideration in the dictionary. After zero-rank LdGSF sequences are removed, the remaining are ordered by rank in line 14. The highest ranked LdGSF sequence is assigned the highest priority because it is viewed as most likely to generate a correct password based on its frequency in the test data. The ordered list is then returned and used to generate a password dictionary.

To build the password dictionary, we defined the $CreateDictionary$ algorithm in Fig-

ure 4.4 with the main focus being on the sets of attributes in each LdGSF. If even one element of the list matches a POI in the given picture, then the LdGSF is beneficial. Otherwise, the entire LdGSF sequence is disregarded. With a valid sequence, a search for all POI combinations that match the sequence attributes are found. The list of combinations are heuristically ordered by pattern as described in line 5.

Each PGA password combination is described as positively horizontal if the gestures placed in the POI locations appear to be in a left-to-right order, negatively horizontal for a right-to-left orientation, positively vertical if the gestures are bottom-to-top, negatively vertical if top-to-bottom, and diagonal if they have both a vertical and horizontal pattern. According to Zhao et al., user gesture patterns are found to be most common in the following order: positively horizontal, diagonal, positively vertical, negatively horizontal, negatively vertical, and the rest follow. These results could not be reproduced in our work therefore, the order of password guesses made by $CreateDictionary$ differed from those in Zhao et al. password dictionary. This heuristically ordered list of applicable sequences derived from LdGSFs is the final password dictionary. The results collected on the number of password guesses may vary based on the assumptions made in designing $CreateDictionary$.

1: **function** CREATEDICTIONARY($order, \theta_k$)
2: **for** $\{\vec{s_1}, \vec{s_2}, \vec{s_3}\}$ in $order$ **do**
3: **for** $\sigma_1, \sigma_2, \sigma_3 \in \theta_k$ **do**
4: **if** $\vec{\sigma_j} \in \vec{s_i}$ **then**
5: $POIlist \leftarrow$ order by Horiz+, Diag, Vert+, Horiz-, Vert- then Other
6: **end if**
7: **end for**
8: **end for**
9: **for** $set \in POIlist$ **do**
10: $dictionary \leftarrow (x_i, y_i) \in set\ \forall i$
11: **end for**
12: **return** $dictionary$
13: **end function**

Figure 4.4. Ordered LdGSFs from Figure 4.3 and an Unseen Picture are Used to Brute Force a Password

Finally, given the algorithms and the data on each picture, we were able to generate password guesses and keep count of how many guesses were made before each password was cracked. These results are analyzed in Chapter 5.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Analysis

In this chapter, we analyze the POIs for the pictures in the Arizona-Turk dataset and discuss the results of the algorithms described in Chapter 4.

5.1 Analyzing Points of Interest

For this research, we analyzed the POIs of the 15 pictures from the Arizona-Turk study, shown in Figure 3.1. Due to PII reasons, we did not have access to the pictures in the Arizona-Student study, so we could not analyze these. Figure 5.1, shows red dashed rectangles on each of the pictures, representing extracted POIs from the images as discussed in Section 4.1. The type of POI is labeled above each rectangle. Each POI is identified as a face, a body, an eye, an ear, a mouth, a nose, a set of head and shoulders, a clock, an airplane, a forehead, or a car. Some POIs were identified only as line, circle, or color type. Other POIs were identified as an unknown objects, or as having no semantics. The algorithm only used the previously listed POIs when creating passwords.

The following are the POIs that were identified for each corresponding picture in Figure 5.1:

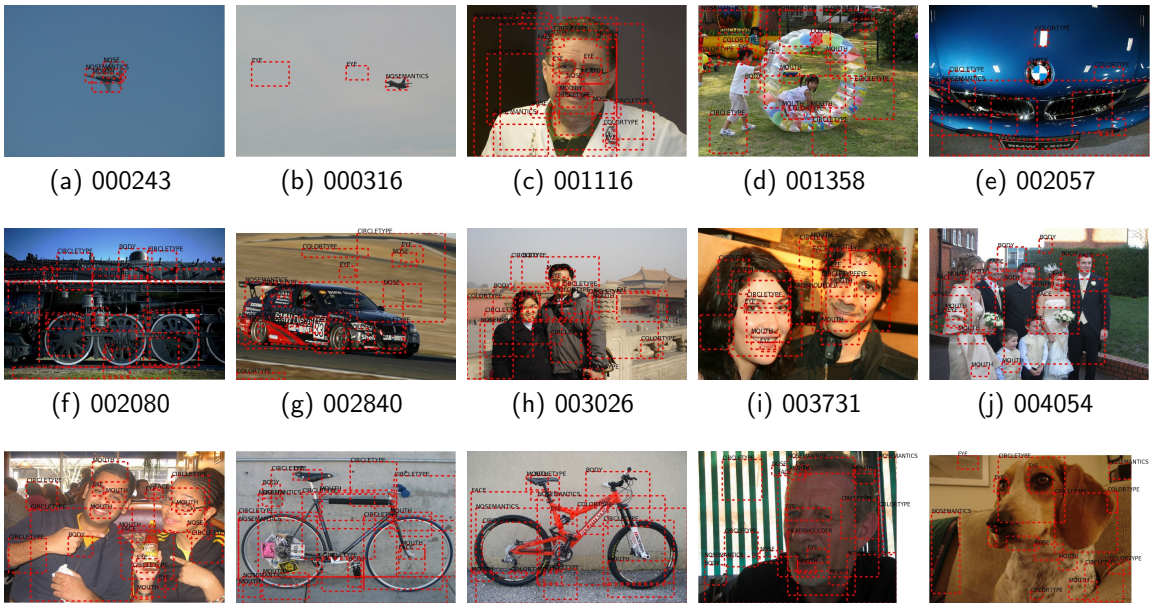
- Figure 5.1(a) is simply a picture of an airplane in the sky, but the POIs identified are a nose, a mouth, and another POI with no semantics.
- Figure 5.1(b) is also an airplane in the sky, yet the POIs identified are two eyes and a POI with no semantics.
- Figure 5.1(c) is a person with the following identified POIs: a body, a face, three eyes, three mouths, three noses, 4 circle types, a color type and 2 POIs with no semantics.
- Figure 5.1(d) is a picture of children playing together with the following POIs identified: 1 body, 6 mouths, 2 eyes, 6 circle types, 4 color types, and one with no semantics.
- Figure 5.1(e) is the front of a BMW automobile. The POIs recognized are a clock, a nose, 5 color types, 3 circle types, and a POI with no semantics.
- Figure 5.1(f) is a close-up picture of a train. The POIs identified are 2 bodies, 7 circle

types, 6 color types, and a POI with no semantics.

- Figure 5.1(g) is a car with the following POIs identified: a face, a mouth, 2 noses, 2 eyes, 4 circle types, 7 color types, and 2 POIs with no semantics.
- Figure 5.1(h) appears to be two tourists standing together. The POIs identified in this picture are 2 bodies, 3 mouths, 2 eyes, 5 circle types, 4 color types, a POI with no semantics.
- Figure 5.1(i) is a picture of a man and a woman. The POIs are 6 mouths, 6 eyes, 2 faces, 5 circle types, and a set of head and shoulders.
- Figure 5.1(j) is a picture with a group of people. The POIs identified are an eye, 5 bodies, 7 faces, and 6 mouths.
- Figure 5.1(k) is another picture of two people with the following identified POIs: a body, a nose, 6 mouths, 2 faces, 3 eyes, and 6 circle types.
- Figure 5.1(l) is a bicycle with the following POIs identified: a body, a face, an eye, 6 mouths, 6 circle types, and 3 POIs with no semantics.
- Figure 5.1(m) is also a bicycle with the following POIs identified: a body, a face, an eye, 4 mouths, 4 circle types, 2 color types, and 2 with no semantics.
- Figure 5.1(n) is a picture of a man. The POIs identified are a body, 2 noses, 3 eyes, 4 mouths, 3 circle types, a color type, a set of head and shoulders, and 4 POIs with no semantics.
- Figure 5.1(o) is a picture of a dog. The POIs found are a nose, 4 eyes, 2 mouths, 3 circle types, 4 color types, and 2 with no semantics.

Clearly, many POIs were incorrectly identified, therefore the source of POI extraction appears not to have been well developed. This led to major consequences when using the $BestCover$ algorithm, which is discussed further in Section 5.2.

Figure 5.2 shows the 15 pictures from the study with their corresponding POI boxes in red and associated passwords in blue. Of note, the password coordinate points tend to fall within the red POI boxes. Specific shapes were used to guide gestures that were made for the passwords, for example heads and wheels were circled, edges had lines associated with them, and eyes were tapped. Any password guess with a single gesture outside the scope of the picture's POIs was not cracked. The algorithm made password guesses based only on information known about the POIs. We did not make password guesses outside the POI boxes shown in red. We did, however, consider circles around POIs, as long as their center



(a) 000243 (b) 000316 (c) 001116 (d) 001358 (e) 002057

(f) 002080 (g) 002840 (h) 003026 (i) 003731 (j) 004054

(k) 005570 (l) 006412 (m) 006467 (n) 007628 (o) 009899

Figure 5.1. Identified POIs of the 15 Pictures from the Arizona-Turk Dataset

point was in a POI. For example, Figures 5.2a and 5.2b are pictures of a small airplane with a clear sky in the background and with the airplane being the only POI in each picture, there were many passwords with gestures made outside of the POI, i.e., in the middle of the sky. Table 5.1 represents the percentage of passwords with either one, two, or all three of their gestures made within POIs, and indicates the chances of the algorithm cracking a password.

This data showed how often users rely on POIs in creating their passwords. For example, in Figure 5.3, by looking only at the passwords for each of these pictures without the background pictures themselves, it is clear that the pictures are bicycles.

5.2 Analyzing *BestCover* Results

Implementing the *BestCover* algorithm (see Section 4.3) on the Arizona-Turk dataset provided the results shown in Figures 5.4 through 5.16. These graphs only show data from passwords that were cracked. The rest of the passwords could not be cracked by the algorithm, therefore, the password guess count is irrelevant.Our results were not comparable to Zhao et al. since their experiments used both the Arizona-Turk dataset and the Arizona-

Table 5.1. Percentage of Passwords Possible to Guess with Number of Gestures in POIs

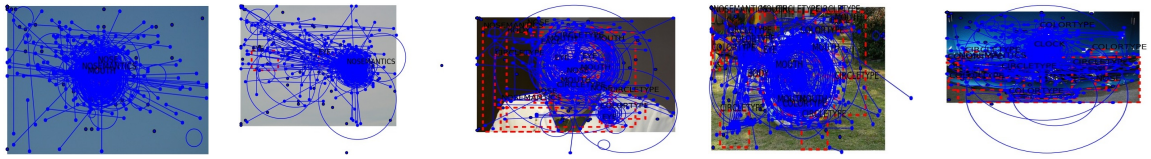
Figure	% passwords with all three gestures outside of POIs	% had exactly two gestures outside of the POIs	% had only one gesture outside of the POIs	% passwords were guessable using algorithm
5.2(a)	19	14	15	52
5.2(b)	20	19	15	46
5.2(c)	3	2	13	82
5.2(d)	1	2	12	82
5.2(e)	7	13	16	86
5.2(f)	4	6	18	72
5.2(g)	4	6	17	73
5.2(h)	6	8	21	65
5.2(i)	0	0	3	97
5.2(j)	4	5	14	76
5.2(k)	3	1	1	85
5.2(l)	0	0	6	94
5.2(m)	3	3	12	82
5.2(n)	0	0	1	99
5.2(o)	5	6	19	70

Student datasets.

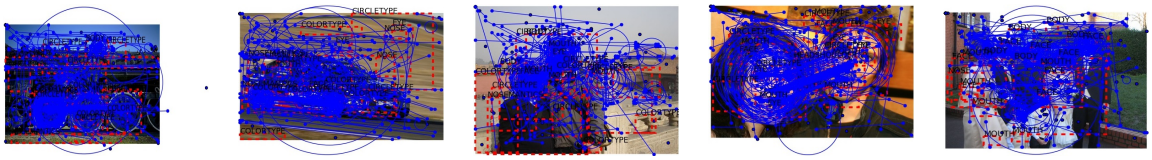
Figure 5.4 shows the results of Figure 3.1(a). As mentioned in Section 5.1, there are very few POIs in this picture, and they were not correctly identified. This made it unlikely that the algorithm would crack the password on this type of picture. Less than 30% of the passwords were cracked, and the uncracked passwords were those with gestures found outside of the POIs. The POIs took up a small area of this picture allowing the algorithm to run quickly. A picture with a minimal amount of POIs should not be used as a background choice.

Figure 5.5 shows the results of Figure 3.1(b). Similar to the last picture, there were very few POIs in this picture, and yet they were all incorrectly identified. Due to the lack of POIs, the algorithm only took a few minutes to run, but only cracked about 30% of the passwords due most gestures being made outside of POIs. Since this picture did not have many POIs, it is not the best choice for a background.

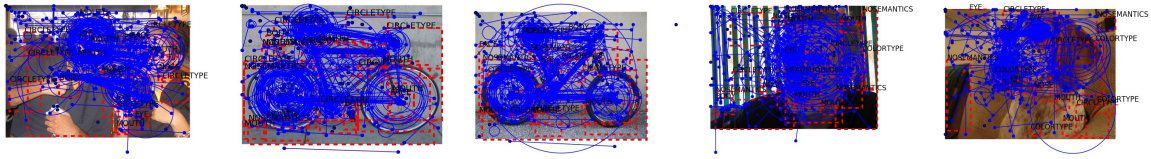
Figure 5.6 shows the results of Figure 3.1(c). There were several POIs, most of which were



(a) 000243 (b) 000316 (c) 001116 (d) 001358 (e) 002057



(f) 002080 (g) 002840 (h) 003026 (i) 003731 (j) 004054



(k) 005570 (l) 006412 (m) 006467 (n) 007628 (o) 009899

Figure 5.2. Passwords of the 15 Pictures from the Arizona-Turk Dataset

(a) 006412 (b) 006467

Figure 5.3. Passwords for Two Pictures of the Arizona-Turk Dataset

accurately identified, and covered most of the area of the picture, allowing the algorithm to crack about 40% of the passwords. Observing the results, we notice that the majority of the passwords were cracked within the same range of guesses. This allows us to think of improvements for the algorithm. Details for improving the algorithm can be found in Section 6.2. Despite the higher password-cracking rate of this picture, this picture is a better background choice compared to the previous ones since it has more POIs, but we will discuss how some of the other pictures are superior choices.

Figure 5.7 shows the results of Figure 3.1(d). The algorithm was able to crack over 30%

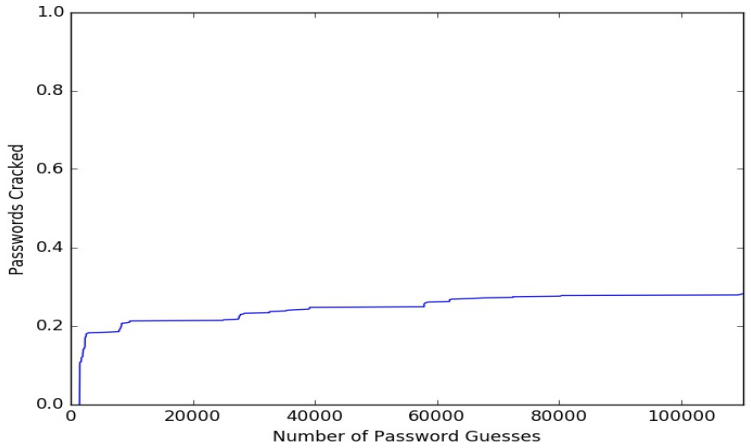


Figure 5.4. CDF Results of Picture 000243.jpg

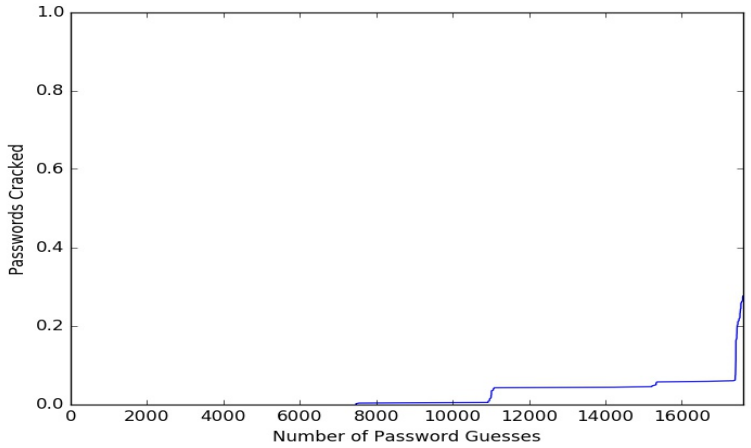


Figure 5.5. CDF Results of Picture 000316.jpg

of the passwords. Observing the results, we notice that about 15% of the passwords were cracked within the same range of guesses. Assuming this jump on the graph was made from the blowup wheel in the picture, the password guesses could have been made sooner with improvements in the algorithm described in Section 6.2. There were unidentified POIs in this background picture that were used as guidance for gestures. Since those POIs were not identified, the algorithm was unable to crack those passwords.

Figure 5.8 shows the results of Figure 3.1(e). The algorithm cracked over 30% of the passwords. We were unable to determine why so many guesses were made before passwords were cracked. It is believed that the overlap caused repeated guesses that should be im-

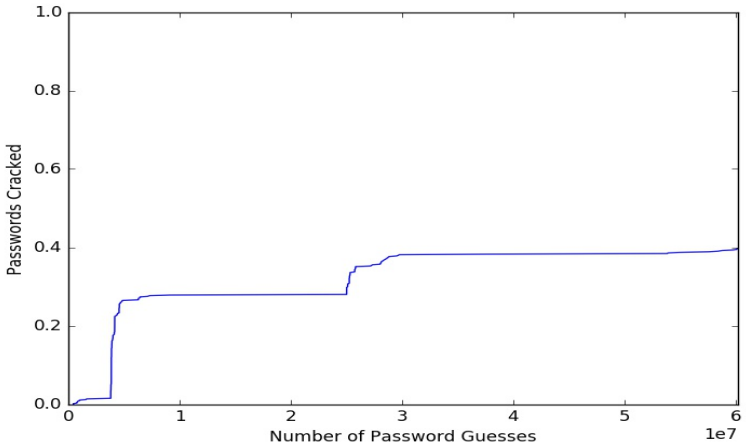


Figure 5.6. CDF Results of Picture 001116.jpg

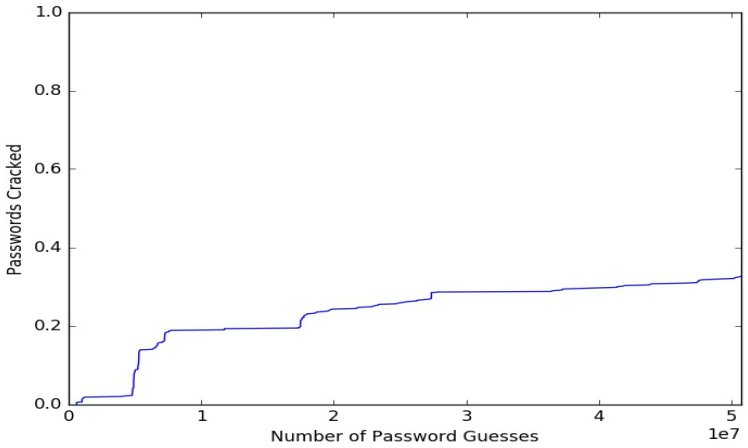


Figure 5.7. CDF Results of Picture 001358.jpg

proved in the algorithm. With the POIs covering only half of the picture and some POIs not identified, this was a stronger picture background.

Figure 5.9 shows the results of Figure 3.1(f). The algorithm cracked about 35% of the passwords. About 2/3 of the passwords cracked were within the same range of guesses. It is safe to assume these passwords that were cracked were the three wheels on the train. This picture is a perfect example to explain how to improve the algorithm to make guesses starting with coordinate points in the midpoint of the POI, instead of bottom-left to the top-right as the algorithm works. More details can be found in Section 6.2. If the wheels were not the main focus of users, this would make a stronger background picture.

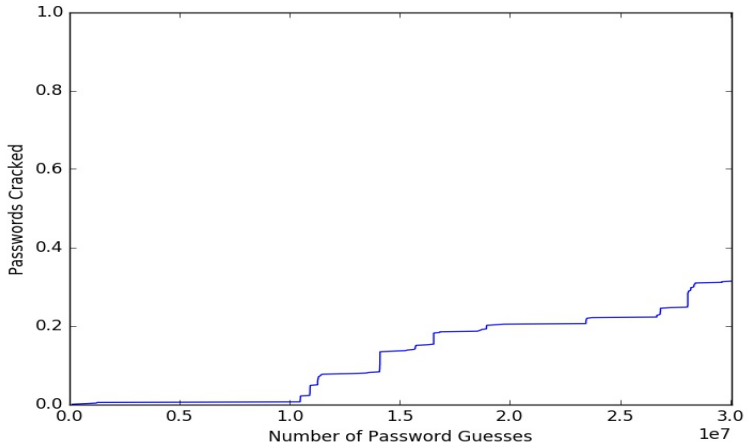


Figure 5.8. CDF Results of Picture 002057.jpg

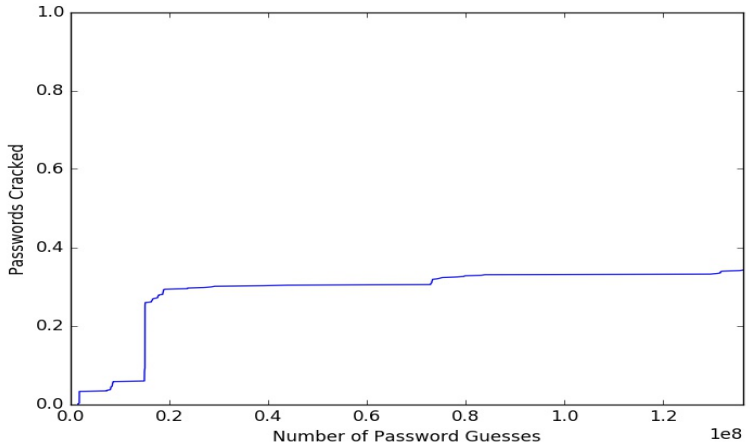


Figure 5.9. CDF Results of Picture 002080.jpg

Figure 5.10 shows the results of Figure 3.1(g). The algorithm cracked over 30% of the passwords. Observing the results, we notice that about 20% of the passwords were quickly cracked. With the entire car identified as a POI, these POIs were able to be cracked.

Figure 5.11 shows the results of Figure 3.1(h). The algorithm cracked about 25% of the passwords. Observing the results, we notice that about 10-15% of the passwords cracked were from circling the heads. Besides those passwords, it was very difficult to crack other passwords with this background since there is so much activity in this picture. This is a great example of a secure background picture.

Figure 5.12 shows the results of Figure 3.1(i). The algorithm cracked over 30% of the

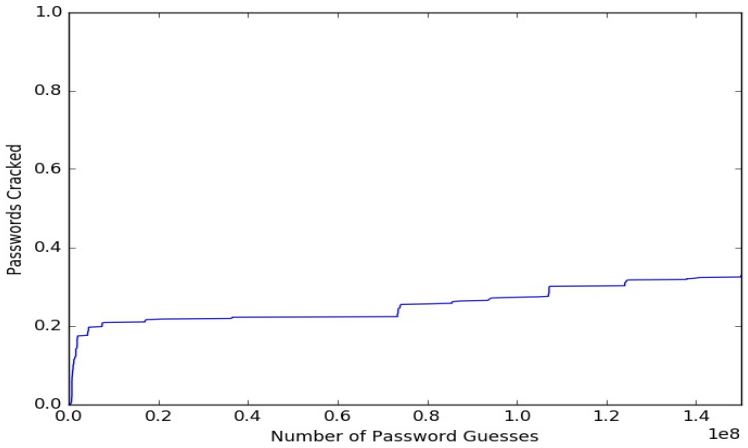


Figure 5.10. CDF Results of Picture 002840.jpg

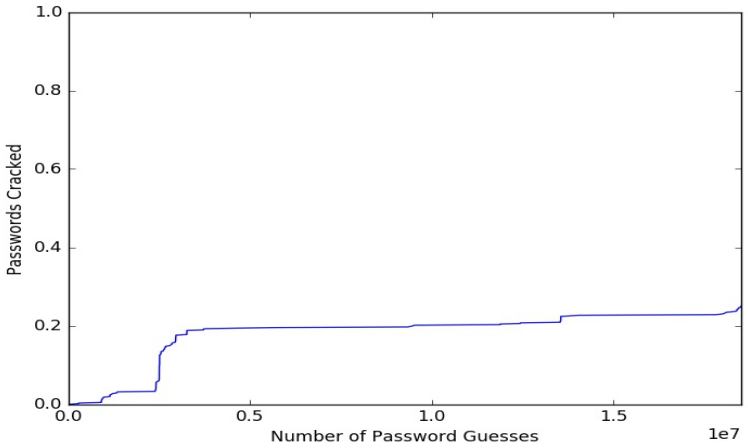


Figure 5.11. CDF Results of Picture 003026.jpg

passwords with 25% of them immediately guessed. A close up picture gives less interesting POIs of interest, making it easy to guess the passwords. Circling heads, tapping eyes, and connecting eyes are the first guesses made. Otherwise, there were not many passwords cracked.

Figure 5.13 shows the results of Figure 3.1(j). The algorithm cracked about 35% of the passwords with 25% of them being a combination of circling heads the heads. If users were using more of a variety of POIs, then there would be significantly fewer passwords cracked.

Figure 5.14 shows the results of Figure 3.1(m). The algorithm cracked just under 30% of

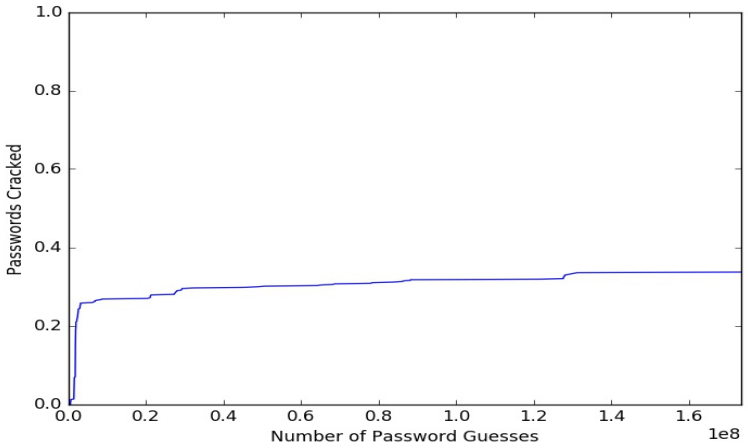


Figure 5.12. CDF Results of Picture 003731.jpg

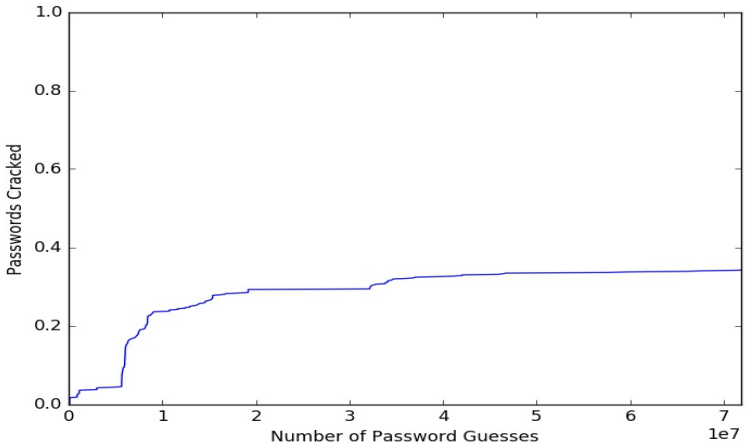


Figure 5.13. CDF Results of Picture 004054.jpg

the passwords. The first 15% of the passwords were using the tires as POIs. Otherwise, the other passwords were difficult to crack. This is a decent background picture since there are many POIs that can be of interest.

Figure 5.15 shows the results of Figure 3.1(n). The algorithm cracked about 40% of the passwords. The first 15% were immediately identified. They must have been in the same class of LdGSFs. With the man's face being the main focus of passwords chosen by users, this is not the best choice of a background picture.

Figure 5.16 shows the results of Figure 3.1(o). The algorithm cracked about 35% of the passwords. About 25% of these passwords were guessed almost simultaneously. Altering

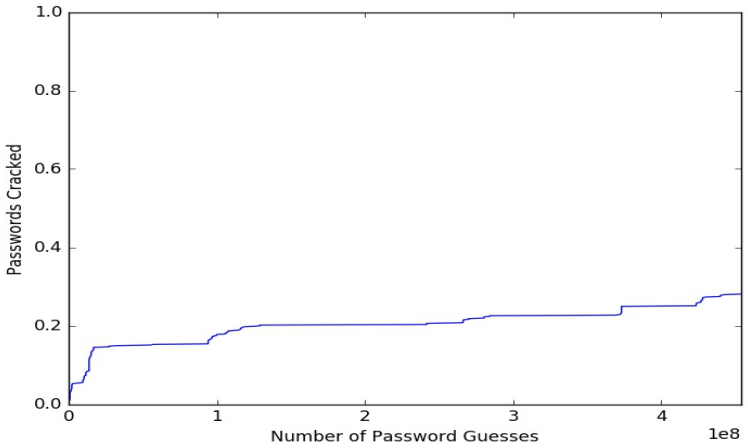


Figure 5.14. CDF Results of Picture 006467.jpg

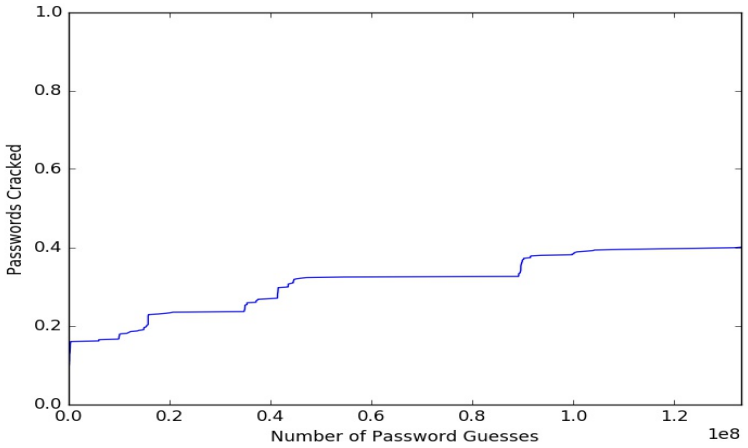


Figure 5.15. CDF Results of Picture 007628.jpg

the program to guess these passwords first would be a major improvement. Not many of the other passwords were cracked. There was not enough of a variety of POIs in this picture for users to vary their passwords, making it a weak background picture.

Depending on the picture used, perhaps because of the number of POIs in the picture, the time taken for the algorithm to break all the passwords varied widely.

5.3 Algorithm Difficulties and Solutions

Our results were not directly comparable to Zhao et al.'s results since the testing and training data used were different, however we were able to create an algorithm that cracks PGA

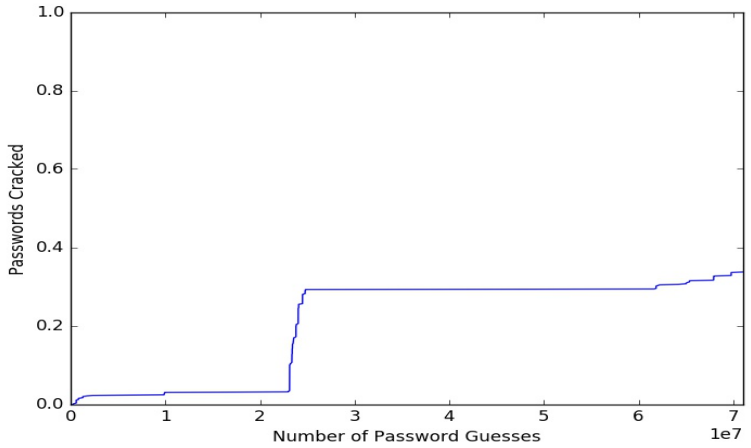


Figure 5.16. CDF Results of Picture 009899.jpg

passwords. Our algorithm used a significant amount of memory, hard disk space, and CPU time to sort and compare the many coordinate points gathered as password guesses, as described in Section 4.3.

Text-based passwords are normally stored using a hash. It is unknown how Microsoft stores PGA passwords but our method described in Chapter 2 used a significant amount of storage. Python[2] dictionary and list data structures were used to keep track of passwords that were cracked and the number of guesses required to crack each password. Suo et al. mentioned that memory storage for password guesses is a difficult problem with PGA [2]. In addition to memory problems, the CPU was not powerful enough on our device to handle the amount of work necessary to run the algorithm.

To address the memory issues and the slow execution on our architecture, we used Amazon Web Services (AWS)[3] to run the algorithms. We created an instance of a c4.xlarge Ubuntu server with 16 GB of memory and 4 CPUs. Due to cost factors, the time spent using the AWS instance was kept to a minimum, roughly $45. The algorithm was run for each of the 15 pictures, on separate CPUs for efficiency.

Even with AWS, however, we were unable to find results for Figures 3.1(k) and 3.1(l) for which the program failed and never completed. There was no error message, such as "MemoryError," to indicate what caused the failures. Attempts to display an exit status

[2]https://www.python.org/
[3]https://aws.amazon.com/

in the terminal (i.e., “echo $?”) also failed. The same results were found after running the program multiple times for each of those pictures. We assume there was possibly an excessive number of passwords generated for these pictures. Perhaps there were far more POIs for these than for the other successful pictures. Fortunately, we achieved results for the latter pictures.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6:
Conclusions and Future Work

In this chapter, we will discuss the accomplishments of this thesis, our recommendations to improve the security of PGA, and future work that can be done to continue the research.

6.1 Conclusions

Each picture from the Arizona-Turk study was investigated in this thesis for its strength as a background picture for PGA. It was found that strong background pictures have a wide variety of POIs. More POIs in a picture implies that there are many more gestures a user can choose from in creating a password. It is assumed that users will choose from among the POIs to assist their choice of password gestures.

An important benefit of this thesis is the creation of a program that can crack gesture passwords. We provided a description on how to crack passwords for PGA. Using data given by Zhao et al., we created visual representations demonstrating the POIs and passwords of the pictures for the Arizona-Turk study. Visuals were created to show efficiency of the program we designed to offer supplementary resources to understand the limits of security of PGA.

Strength requirements for PGA passwords, just as there are for text-based passwords, will improve the security of PGA. For example, strength requirements for Windows 8 and Windows 10 might be to increase the number of gestures per password, add new types of gestures, and ensure the picture chosen by the user contains numerous POIs dispersed across the picture. Using a smaller error distance, as discussed in Section 2.1, will force an attacker to make more guesses, however this can cause false negatives when valid users attempt to log in. Until such strength requirements are available, we conclude that it would be beneficial to use a different means of authentication for the security of government information.

6.2 Future Work

We have developed a working program that produces sensible guesses to crack PGA passwords. Ideally, this program can be improved in the following ways:

- Most importantly, the algorithm can be enhanced by making fewer guesses.
- Advancements can also be accomplished by refining memory issues and increasing speed. This can be done by using a better POI detection program, and considering programming languages other than Python.
- Since the coordinate points guessed in the algorithm are made in order from the bottom-left to the top-right, an improvement might be to randomize the order of password guesses in the list of guesses made for each heuristically ordered pattern, as described in Section 4.3.
- Another solution to the same problem may be to begin at the center of each POI, which would "hit" the commonly used midpoints of the circle.
- Furthermore, the algorithm can be designed to construct password guesses outside of the POIs in the picture, but at that point, it would be brute-forcing.
- Finally, it is intended that the program works for unseen pictures. This may be utilized by adding an algorithm that locates POIs and records the coordinate locations of the POIs. With this information, the brute-force algorithm in Chapter 4 can guess passwords for unseen pictures.

List of References

[1] J. Eaton, "The political significance of the imperial watchword in the early empire," *Greece and Rome (Second Series)*, vol. 58, no. 01, pp. 48–63, 2011.

[2] X. Suo, Y. Zhu, and G. S. Owen, "Graphical passwords: A survey," in *21st Annual Computer Security Applications Conference (ACSAC'05)*. IEEE, 2005, pp. 10.

[3] Z. Zhao, G.-J. Ahn, and H. Hu, "Picture gesture authentication: Empirical analysis, automated attacks, and scheme evaluation," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 4, p. 14, 2015.

[4] Z. Zhao, G.-J. Ahn, J.-J. Seo, and H. Hu, "On the security of picture gesture authentication." in *USENIX Security*, 2013, pp. 383–398.

[5] S. Sinofsky. (2011, Dec. 16). Signing in with a picture password. [Online]. Available: https://blogs.msdn.microsoft.com/b8/2011/12/16/signing-in-with-a-picture-password/

[6] "DOD cybersecurity discipline implementation plan." Available: http://dodcio.defense.gov/Portals/0/Documents/Cyber/CyberDis-ImpPlan.pdf

[7] D. Damopoulos, G. Kambourakis, and S. Gritzalis, "From keyloggers to touchloggers: Take the rough with the smooth," *Computers & Security*, vol. 32, pp. 102–114, 2013.

[8] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and I know it's you!: Implicit authentication based on touch screen patterns," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 987–996.

[9] J. Thorpe and P. C. van Oorschot, "Graphical dictionaries and the memorable space of graphical passwords," in *USENIX Security Symposium*, 2004, pp. 135–150.

[10] D. Nali and J. Thorpe, "Analyzing user choice in graphical passwords," *School of Computer Science, Carleton University, Tech. Rep. TR-04-01*, 2004.

[11] P. Dunphy and J. Yan, "Do background images improve draw a secret graphical passwords?" in *Proceedings of the 14th ACM Conference on Computer and Communications Security*. ACM, 2007, pp. 36–47.

[12] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "Passpoints: Design and longitudinal evaluation of a graphical password system," *International Journal of Human-Computer Studies*, vol. 63, no. 1, pp. 102–127, 2005.

[13] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "Authentication using graphical passwords: effects of tolerance and image choice," in *Proceedings of the 2005 symposium on Usable privacy and security*. ACM, 2005, pp. 1–12.

[14] M. N. Alshehri, H. Crawford, and L. M. Mayron. (2015). Poster: Image suitability for graphical passwords. IEEE. [Online]. pp. 2. Available: http://www.ieee-security.org/TC/SP2015/posters/paper_4.pdf

[15] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens." *WOOT*, vol. 10, pp. 1–7, 2010.

[16] J. Maderer. (2014, Apr. 7) Personal touch signature makes mobile devices more secure. [Online]. Available: http://www.news.gatech.edu/2014/04/07/personal-touch-signature-makes-mobile-devices-more-secure

[17] P. Saravanan, S. Clarke, D. H. P. Chau, and H. Zha, "Latentgesture: Active user authentication through background touch analysis," in *Proceedings of the Second International Symposium of Chinese CHI*. ACM, 2014, pp. 110–113.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2007, Apr.). The pascal visual object classes challenge 2007 voc2007 results. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California